AMENDMENTS to the CLAIMS:

Please amend the claims as follows:

1. (original) A method in a data processing system having a dependent node, a defining node, and a plurality of dependencies between the dependent node and the defining node, the method comprising the steps of:

displaying a graphical representation of the dependent node;

displaying a graphical representation of the defining node;

receiving an indication to identify a dependency between the dependent node and the defining node; and

in response to receiving the indication to identify the dependency, representing the plurality of dependencies as a number of links that is less than a number of the dependencies between the dependent node and the defining node.

2. (original) The method of claim 1, wherein the plurality of dependencies is represented as a single link.

3. (original) The method of claim 1, wherein each node comprises an element.

4. (original) The method of claim 3, wherein one of the plurality of dependencies comprises a use of the defining node element by the dependent node element.

5. (original) The method of claim 3, wherein one of the plurality of dependencies comprises a declaration of the defining node element by the dependent node element.

6. (original) The method of claim 3, wherein one of the plurality of dependencies comprises a call to a method of the defining node element by the dependent node element.

7. (original) The method of claim 3, wherein one of the plurality of dependencies comprises a local variable definition using the defining node element in a method of the dependent node element.

8. (original) The method of claim 3, wherein the dependent node element comprises a class.

9. (original) The method of claim 3, wherein the dependent node element comprises an interface.

10. (original) The method of claim 3, wherein the defining node element comprises a class.

11. (original) The method of claim 3, wherein the defining node element comprises an interface.

12. (currently amended) The method of claim 1, wherein the dependent node ~~comprises~~ is selected from the group consisting of a class, an interface, and a package.

13 - 14.(cancelled)

15. (currently amended) The method of claim ~~14~~ 12, wherein the package comprises a plurality of elements.

16. (original) The method of claim 15, wherein one of the plurality of elements comprises a class.

17. (original) The method of claim 15, wherein one of the plurality of elements comprises an interface.

18. (currently amended) The method of claim 1, wherein the defining node ~~comprises~~ is selected from the group consisting of a class, an interface, and a package.

19 - 20. (cancelled)

21. (currently amended) The method of claim ~~20~~ 18, wherein the package comprises a plurality of elements.

22. (original) The method of claim 21, wherein one of the plurality of elements comprises a class.

23. (original) The method of claim 21, wherein one of the plurality of elements comprises an interface.

24. (original) A method in a data processing system having a plurality of nodes, each of the plurality of nodes having corresponding code, the method comprising the steps of:

displaying a graphical representation of the plurality of nodes;

determining whether the code corresponding to a first of the plurality of nodes contains a first use of a second of the plurality of nodes; and

when it is determined that the code corresponding to the first node contains the first use of the second node, determining whether the code corresponding to the first node contains a second use of the second node; and

when it is determined that the code corresponding to the first node contains the second use of the second node, displaying a dependency link between the graphical representation of the first node and the graphical representation of the second node.

25. (original) The method of claim 24, wherein the first use comprises a declaration.

26. (original) The method of claim 25, wherein the step of determining whether the code corresponding to the first node contains the first use of the second node comprises the step of searching the code corresponding to the first node for an attribute declaration that uses the second node.

27. (original) The method of claim 25, wherein the step of determining whether the code corresponding to the first node contains the first use of the second node comprises the step of searching the code corresponding to the first node for an initializer of an attribute declaration that uses the second node.

28. (original) The method of claim 25, wherein the step of determining whether the code corresponding to the first node contains the first use of the second node comprises the step of searching the code corresponding to the first node for an argument parameter of a method that uses the second node.

29. (currently amended) The method of claim 24, wherein the first use of the second node comprises a call to a method ~~call~~ of the second node.

30. (original) The method of claim 24, wherein the first use of the second node comprises a local variable definition using the second node in a method of the first node.

31. (currently amended) The method of claim 24, wherein the first node ~~comprises~~ is selected from the group consisting of a class, an interface, and a package.

32 - 33. (cancelled)

34. (currently amended) The method of claim 24, wherein the second node ~~comprises~~ is selected from the group consisting of a class, an interface, and a package.

35 - 37. (cancelled)

38. (currently amended) The method of claim ~~37~~ 226, ~~wherein~~ further comprising the steps of:

when it is determined that the code corresponding to the first node does not contain the ~~other~~ second declaration of the second node, determining

whether the code corresponding to the first node contains a call to a method of the second node; and

when it is determined that the code corresponding to the first node contains the call to the method of the second node, displaying a dependency link between the graphical representation of the first node and the graphical representation of the second node.

39. (currently amended) The method of claim ~~37~~ 226, ~~wherein~~ further comprising the steps of:

when it is determined that the code corresponding to the first node does not contain the ~~other~~ second declaration of the second node, determining whether the code corresponding to the first node comprises a method having a local variable definition using the second node; and

when it is determined that the code corresponding to the first node comprises a method having the local variable definition using the second node, displaying a dependency link between the graphical representation of the first node and the graphical representation of the second node.

40 - 46. (cancelled)

47. (currently amended) The method of claim ~~46~~ 227, ~~wherein~~ further comprising the steps of:

when it is determined that the code corresponding to the first node does not contain the ~~other~~ second call to the method of the second node, determining whether the code corresponding to the first node comprises a method having a local variable definition using the second node; and

when it is determined that the code corresponding to the first node comprises a method having the local variable definition using the second node, displaying a dependency link between the graphical representation of the first node and the graphical representation of the second node.

48. (currently amended) The method of claim ~~46~~ 227, ~~wherein~~ further comprising the steps of:

when it is determined that the code corresponding to the first node does not contain the ~~other~~ second call to the method of the second node, determining whether the code corresponding to the first node contains a declaration of the second node; and

when it is determined that the code corresponding to the first node contains the declaration of the second node, displaying a dependency link between the graphical representation of the first node and the graphical representation of the second node.

49 - 55. (cancelled)

56. (currently amended) The method of claim ~~55~~ 65, wherein the first use comprises a declaration.

57. (currently amended) The method of claim ~~55~~ 65, wherein the first use comprises a ~~method~~ call to a method of the second node.

58. (currently amended) The method of claim ~~55~~ 65, wherein the first use of the second node comprises a local variable definition using the second node in a method of the first node.

59 - 64. (cancelled)

65. (currently amended) A method in a data processing system having a plurality of nodes, each of the plurality of nodes having corresponding code, the method comprising the steps of:

receiving an indication of a first of the plurality of nodes;

receiving an indication of a second of the plurality of nodes;

determining whether the code corresponding to the first node contains a ~~declaration~~ first use of the second node; and

when it is determined that the code corresponding to the first node contains the ~~declaration~~ first use of the second node, determining whether the code corresponding to the first node contains ~~another declaration~~ a second use of the second node; and

when it is determined that the code corresponding to the first node contains the ~~other declaration~~ second use of the second node, displaying a dependency link between the graphical representation of the first node and the graphical representation of the second node.

66. (currently amended) The method of claim 65, wherein the first node ~~comprises~~ is selected from the group consisting of a class, an interface, and a package.

67 -68. (cancelled)

69. (currently amended) The method of claim 65, wherein the second node ~~comprises~~ is selected from the group consisting of a class, an interface, and a package.

70 - 78. (cancelled)

79. (original) A method in a data processing system having a plurality of nodes, each of the plurality of nodes having corresponding code, the method comprising the steps of:

receiving an indication of a first of the plurality of nodes;

receiving an indication of a second of the plurality of nodes;

determining whether the code corresponding to the first node contains a use of the second node; and

when it is determined that the code corresponding to the first node contains the use of the second node, displaying the usage of the second node by the first node.

80. (original) The method of claim 79, wherein each node comprises an element.

81. (currently amended) The method of claim ~~79~~ 80, wherein when it is determined that the code corresponding to the first node contains the use of the second node, the method further comprises the step of displaying the first node element.

82. (original) The method of claim 81, further comprising the step of displaying the first node with the first node element and with the usage to visually indicate that the first node contains the usage of the second node.

83. (currently amended) The method of claim ~~79~~ 80, wherein when it is determined that the code corresponding to the first node contains the use of the second node, the method further comprises the step of displaying the second node element.

84. (original) The method of claim 83, further comprising the step of displaying the second node with the second node element and with the usage to visually indicate that the first node contains the usage of the second node.

85. (currently amended) The method of claim 79, wherein the first node ~~comprises~~ is selected from the group consisting of a class, an interface, and a package.

86 - 87. (cancelled)

88. (currently amended) The method of claim 79, wherein the second node ~~comprises~~ is selected from the group consisting of a class, an interface, and a package.

89 - 90. (cancelled)

91. (original) A computer-readable medium containing instructions for controlling a data processing system to perform a method, the data processing system having a dependent node, a defining node, and a plurality of dependencies between the dependent node and the defining node, the method comprising the steps of:

displaying a graphical representation of the dependent node;

displaying a graphical representation of the defining node;

receiving an indication to identify a dependency between the dependent node and the defining node; and

in response to receiving the indication to identify the dependency, representing the plurality of dependencies as a number of links that is less than a number of the dependencies between the dependent node and the defining node.

92. (original) The computer-readable medium of claim 91, wherein the plurality of dependencies is represented as a single link.

93. (original) The computer-readable medium of claim 91, wherein each node comprises an element.

94. (original) The computer-readable medium of claim 93, wherein one of the plurality of dependencies comprises a use of the defining node element by the dependent node element.

95. (original) The computer-readable medium of claim 93, wherein one of the plurality of dependencies comprises a declaration of the defining node element by the dependent node element.

96. (original) The computer-readable medium of claim 93, wherein one of the plurality of dependencies comprises a call to a method of the defining node element by the dependent node element.

97. (original) The computer-readable medium of claim 93, wherein one of the plurality of dependencies comprises a local variable definition using the defining node element in a method of the dependent node element.

98. (original) The computer-readable medium of claim 93, wherein the dependent node element comprises a class.

99. (original) The computer-readable medium of claim 93, wherein the dependent node element comprises an interface.

100. (original) The computer-readable medium of claim 93, wherein the defining node element comprises a class.

101. (original) The computer-readable medium of claim 93, wherein the defining node element comprises an interface.

102. (currently amended) The computer-readable medium of claim 91, wherein the dependent node ~~comprises~~ is selected from the group consisting of a class, an interface and a package.

103 - 104. (cancelled)

105. (currently amended) The computer-readable medium of claim ~~104~~ 102, wherein the package comprises a plurality of elements.

106. (original) The computer-readable medium of claim 105, wherein one of the plurality of elements comprises a class.

107. (original) The computer-readable medium of claim 105, wherein one of the plurality of elements comprises an interface.

108. (currently amended) The computer-readable medium of claim 91, wherein the defining node ~~comprises~~ is selected from the group consisting of a class, an interface and a package.

109 - 110. (cancelled)

111. (currently amended) The computer-readable medium of claim ~~110~~ 108, wherein the package comprises a plurality of elements.

112. (original) The computer-readable medium of claim 111, wherein one of the plurality of elements comprises a class.

113. (original) The computer-readable medium of claim 111, wherein one of the plurality of elements comprises an interface.

114. (original) A computer-readable medium containing instructions for controlling a data processing system to perform a method, the data processing system having a plurality of nodes, each of the plurality of nodes having corresponding code, the method comprising the steps of:

displaying a graphical representation of the plurality of nodes;

determining whether the code corresponding to a first of the plurality of nodes contains a first use of a second of the plurality of nodes; and

when it is determined that the code corresponding to the first node contains the first use of the second node, determining whether the code corresponding to the first node contains a second use of the second node; and

when it is determined that the code corresponding to the first node contains the second use of the second node, displaying a dependency link between the graphical representation of the first node and the graphical representation of the second node.

115. (original) The computer-readable medium of claim 114, wherein the first use comprises a declaration.

116. (original) The computer-readable medium of claim 115, wherein the step of determining whether the code corresponding to the first node contains the first use of the second node comprises the step of searching the code corresponding to the first node for an attribute declaration that uses the second node.

117. (original) The computer-readable medium of claim 115, wherein the step of determining whether the code corresponding to the first node contains the first use of the second node comprises the step of searching the code corresponding to the first node for an initializer of an attribute declaration that uses the second node.

118. (original) The computer-readable medium of claim 115, wherein the step of determining whether the code corresponding to the first node contains the first use of the second node comprises the step of searching the code corresponding to the first node for an argument parameter of a method that uses the second node.

119. (currently amended) The computer-readable medium of claim 114, wherein the first use of the second node comprises a call to a method ~~call~~ of the second node.

120. (original) The computer-readable medium of claim 114, wherein the first use of the second node comprises a local variable definition using the second node in a method of the first node.

121. (currently amended) The computer-readable medium of claim 114, wherein the first node ~~comprises~~ is selected from the group consisting of a class, an interface, and a package.

122 - 123. (cancelled)

124. (currently amended) The computer-readable medium of claim 114, wherein the second node ~~comprises~~ is selected from the group consisting of a class, an interface, and a package.

125 - 127. (cancelled)

128. (currently amended) The computer-readable medium of claim ~~127~~ 235, wherein when it is determined that the code corresponding to the first node does not contain the ~~other~~ second declaration of the second node, the method further comprises the steps of:

determining whether the code corresponding to the first node contains a call to a method of the second node; and

when it is determined that the code corresponding to the first node contains the call to the method of the second node, displaying a dependency link between the graphical representation of the first node and the graphical representation of the second node.

129. (currently amended) The computer-readable medium of claim ~~127~~ 235, wherein when it is determined that the code corresponding to the first node does not contain the ~~other~~ second declaration of the second node, the method further comprises the steps of:

determining whether the code corresponding to the first node comprises a method having a local variable definition using the second node; and

when it is determined that the code corresponding to the first node comprises a method having the local variable definition using the second node, displaying a dependency link between the graphical representation of the first node and the graphical representation of the second node.

130 - 136. (cancelled)

137. (currently amended) The computer-readable medium of claim ~~136~~ 236, wherein when it is determined that the code corresponding to the first node does not contain the ~~other~~ second call to the method of the second node, the method further comprises the steps of:

determining whether the code corresponding to the first node comprises a method having a local variable definition using the second node; and

when it is determined that the code corresponding to the first node comprises a method having the local variable definition using the second node, displaying a dependency link between the graphical representation of the first node and the graphical representation of the second node.

138. (currently amended) The computer-readable medium of claim ~~136~~ 236, wherein when it is determined that the code corresponding to the first node does not contain the ~~other~~ second call to the method of the second node, the method further comprises the steps of:

determining whether the code corresponding to the first node contains a declaration of the second node; and

when it is determined that the code corresponding to the first node contains the declaration of the second node, displaying a dependency link between the graphical representation of the first node and the graphical representation of the second node.

139 - 145. (cancelled)

146. (currently amended) The computer-readable medium of claim ~~145~~ 155, wherein the first use comprises a declaration.

147. (currently amended) The computer-readable medium of claim ~~145~~ 155, wherein the first use of the second node comprises a ~~method~~ call to a method of the second node.

148. (currently amended) The computer-readable medium of claim ~~145~~ 155, wherein the first use of the second node comprises a local variable definition using the second node in a method of the first node.

149 - 154. (cancelled)

155. (currently amended) A computer-readable medium containing instructions for controlling a data processing system to perform a method, the data processing system having a plurality of nodes, each of the plurality of nodes having corresponding code, the method comprising the steps of:

receiving an indication of a first of the plurality of nodes;

receiving an indication of a second of the plurality of nodes;

determining whether the code corresponding to the first node contains a ~~declaration~~ first use of the second node; and

when it is determined that the code corresponding to the first node contains the ~~declaration~~ first use of the second node, determining whether the code

corresponding to the first node contains ~~another declaration~~ a second use of the second node; and

when it is determined that the code corresponding to the first node contains the ~~other declaration~~ second use of the second node, displaying a dependency link between the graphical representation of the first node and the graphical representation of the second node.

156. (currently amended) The computer-readable medium of claim 155, wherein the first node ~~comprises~~ is selected from the group consisting of a class, an interface, and a package.

157 - 158. (cancelled)

159. (currently amended) The computer-readable medium of claim 155, wherein the second node ~~comprises~~ is selected from the group consisting of a class, an interface, and a package.

160 - 168. (cancelled)

169. (original) A computer-readable medium containing instructions for controlling a data processing system to perform a method, the data processing system having a plurality of nodes, each of the plurality of nodes having corresponding code, the method comprising the steps of:

receiving an indication of a first of the plurality of nodes;

receiving an indication of a second of the plurality of nodes;

determining whether the code corresponding to the first node contains a use of the second node; and

when it is determined that the code corresponding to the first node contains the use of the second node, displaying the usage of the second node by the first node.

170. (original) The computer-readable medium of claim 169, wherein each node comprises an element.

171. (currently amended) The computer-readable medium of claim ~~169~~ 170, wherein when it is determined that the code corresponding to the first node contains the use of the second node, the method further comprises the step of displaying the first node element.

172. (original) The computer-readable medium of claim 171, wherein the method further comprises the step of displaying the first node with the first node element and with the usage to visually indicate that the first node contains the usage of the second node.

173. (currently amended) The computer-readable medium of claim ~~169~~ 170, wherein when it is determined that the code corresponding to the first node contains the use of the second node, the method further comprises the step of displaying the second node element.

174. (original) The computer-readable medium of claim 173, wherein the method further comprises the step of displaying the second node with the second node element and with the usage to visually indicate that the first node contains the usage of the second node.

175. (currently amended) The computer-readable medium of claim 169, wherein the first node ~~comprises~~ is selected from the group consisting of a class, an interface, and a package.

176 - 177. (cancelled)

178. (currently amended) The computer-readable medium of claim 169, wherein the second node ~~comprises~~ is selected from the group consisting of a class, an interface, and a package.

179 - 180. (cancelled)

181. (currently amended) A data processing system comprising:

a secondary storage device further comprising a plurality of nodes, each of the plurality of nodes having corresponding code;

a memory device further comprising a program that displays a graphical representation of the plurality of nodes, that

determines whether the code corresponding to a first of the plurality of nodes contains a ~~declaration~~ first use of ~~the~~ a second of the plurality of nodes, and

when it is determined that the code corresponding to the first node contains the ~~declaration~~ first use of the second node, the program determines whether the code corresponding to the first node contains ~~another declaration~~ a second use of the second node, and

when it is determined that the code corresponding to the first node contains the ~~other declaration~~ second use of the second node, the program displays a dependency link between the graphical representation of the first node and the graphical representation of the second node; and

a processor for running the program.

182. (currently amended) The data processing system of claim ~~181~~ 245, wherein when it is determined that the code corresponding to the first node does not contain the ~~other~~ second declaration of the second node, the program

determines whether the code corresponding to the first node contains a call to a method of the second node, and

when it is determined that the code corresponding to the first node contains the call to the method of the second node, the program displays a dependency link between the graphical representation of the first node and the graphical representation of the second node.

183. (currently amended) The data processing system of claim ~~181~~ 245, wherein when it is determined that the code corresponding to the first node does not contain the ~~other~~ second declaration of the second node, the program

determines whether the code corresponding to the first node comprises a method having a local variable definition using the second node, and

when it is determined that the code corresponding to the first node comprises a method having the local variable definition using the second node, the program displays a dependency link between the graphical representation of the first node and the graphical representation of the second node.

184. (currently amended) The data processing system of claim 181, wherein the first node ~~comprises~~ is selected from the group comprising a class, and interface, and a package.

185 - 186. (cancelled)

187. (currently amended) The data processing system of claim 181, wherein the second node ~~comprises~~ is selected from the group comprising a class, an interface, and a package.

188 - 190. (cancelled)

191. (currently amended) The data processing system of claim ~~190~~ 247, wherein when it is determined that the code corresponding to the first node does not contain the ~~other~~ second call to the method of the second node, the program

determines whether the code corresponding to the first node comprises a method having a local variable definition using the second node, and

when it is determined that the code corresponding to the first node comprises a method having the local variable definition using the second node, the program displays a dependency link between the graphical representation of the first node and the graphical representation of the second node.

192. (currently amended) The data processing system of claim ~~190~~ 247, wherein when it is determined that the code corresponding to the first node does not contain the ~~other~~ second call to the method of the second node, the program

determines whether the code corresponding to the first node contains a declaration of the second node, and

when it is determined that the code corresponding to the first node contains the declaration of the second node, the program displays a dependency link between the graphical representation of the first node and the graphical representation of the second node.

193 - 198. (cancelled)

199. (currently amended) A data processing system comprising:

a secondary storage device further comprising a plurality of nodes, each of the plurality of nodes having corresponding code;

a memory device further comprising a program that receives an indication of a first of the plurality of nodes, that

receives an indication of a second of the plurality of nodes, that determines whether the code corresponding to the first node contains a ~~declaration~~ first use of the second node, and

when it is determined that the code corresponding to the first node contains the ~~declaration~~ first use of the second node, the program determines whether the code corresponding to the first node contains ~~another declaration~~ a second use of the second node, and

when it is determined that the code corresponding to the first node contains the ~~other declaration~~ second use of the second node, the program displays a dependency link between the graphical representation of the first node and the graphical representation of the second node; and

a processor for running the program.

200. (currently amended) The data processing system of claim 199, wherein the first node ~~comprises~~ is selected from the group consisting of a class, an interface, and a package.

201 - 202. (cancelled)

203. (currently amended) The data processing system of claim 199, wherein the second node ~~comprises~~ is selected from the group consisting of a class, an interface, and a package.

204 - 212. (cancelled)

213. (original) A data processing system comprising:

a secondary storage device further comprising a plurality of nodes, each of the plurality of nodes having corresponding code;

a memory device further comprising a program that receives an indication of a first of the plurality of nodes, that

receives an indication of a second of the plurality of nodes, that

determines whether the code corresponding to the first node contains a use of the second node, and

when it is determined that the code corresponding to the first node contains the use of the second node, the program displays the usage of the second node by the first node; and

a processor for running the program.

214. (original) The data processing system of claim 213, wherein each node comprises an element.

215. (currently amended) The data processing system of claim ~~213~~ 214, wherein when it is determined that the code corresponding to the first node contains the use of the second node, the program further displays the first node element.

216. (original) The data processing system of claim 215, wherein the program further displays the first node with the first node element and with the usage to visually indicate that the first node contains the usage of the second node.

217. (currently amended) The data processing system of claim ~~213~~ 214, wherein when it is determined that the code corresponding to the first node contains the use of the second node, the program further displays the second node element.

218. (original) The data processing system of claim 217, wherein the program further displays the second node with the second node element and with the usage to visually indicate that the first node contains the usage of the second node.

219. (currently amended) The data processing system of claim 213, wherein the first node ~~comprises~~ is selected from the group comprising a class, an interface, and a package.

220 - 221. (cancelled)

222. (currently amended) The data processing system of claim 213, wherein the second node ~~comprises~~ is selected from the group comprising a class, an interface, and a package.

223 - 224. (cancelled)

225. (original) A system having a dependent node, a defining node, and a plurality of dependencies between the dependent node and the defining node, the system comprising:

means for displaying a graphical representation of the dependent node;

means for displaying a graphical representation of the defining node;

means for receiving an indication to identify a dependency between the dependent node and the defining node; and

means for representing the plurality of dependencies as a number of links that is less than a number of the dependencies between the dependent node and the defining node in response to receiving the indication to identify the dependency.

226. (new) The method of claim 25, wherein the second use comprises a second declaration.

227. (new) The method of claim 29, wherein the second use of the second node comprises a second call to the method of the second node.

228. (new) The method of claim 30, wherein the second use of the second node comprises a local variable definition using the second node in a method of the first node.

229. (new) The method of claim 56, wherein the second use comprises a second declaration.

230. (new) The method of claim 229, further comprising the steps of :

when it is determined that the code corresponding to the first node does not contain the second declaration of the second node, determining whether the code corresponding to the first node contains a call to a method of the second node; and

when it is determined that the code corresponding to the first node contains the call to the method of the second node, displaying a dependency link between the graphical representation of the first node and the graphical representation of the second node.

231. (new) The method of claim 229, further comprising the steps of:

when it is determined that the code corresponding to the first node does not contain the second declaration of the second node, determining whether the code corresponding to the first node comprises a method having a local variable definition using the second node; and

when it is determined that the code corresponding to the first node comprises the method having the local variable definition using the second node, displaying a dependency link between the graphical representation of the first node and the graphical representation of the second node.

232. (new) The method of claim 57, wherein the second use comprises a call to the method of the second node.

233. (new) The method of claim 57, further comprising the steps of:

when it is determined that the code corresponding to the first node does not contain a second call to the method of the second node, determining whether the code corresponding to the first node comprises a method having a local variable definition using the second node; and

when it is determined that the code corresponding to the first node comprises a method having the local variable definition using the second node, displaying a dependency link between the graphical representation of the first node and the graphical representation of the second node.

234. (new) The method of claim 58, wherein the second use of the second node comprises a local variable definition using the second node in a method of the first node.

235. (new) The computer-readable medium of claim 115, wherein the second use comprises a second declaration.

236. (new) The computer-readable medium of claim 119, wherein the second use of the second node comprises a second call to the method of the second node.

237. (new) The computer-readable medium of claim 120, wherein the second use of the second node comprises a second local variable definition using the second node in a method of the first node.

238. (new) The computer-readable medium of claim 146, wherein the second use comprises a second declaration.

239. (new) The computer-readable medium of claim 238, wherein:

when it is determined that the code corresponding to the first node does not contain the second declaration of the second node, determining whether the code corresponding to the first node contains a call to a method of the second node; and

when it is determined that the code corresponding to the first node contains the call to the method of the second node, displaying a dependency link between the graphical representation of the first node and the graphical representation of the second node.

240. (new) The computer readable medium of claim 238, wherein:

when it is determined that the code corresponding to the first node does not contain the second declaration of the second node, determining whether the code corresponding to the first node comprises a method having a local variable definition using the second node; and

when it is determined that the code corresponding to the first node comprises a method having the local variable definition using the second node,

displaying a dependency link between the graphical representation of the first node and the graphical representation of the second node.

241. (new) The computer-readable medium of claim 147, wherein the second use comprises a second call to the method of the second node.

242. (new) The computer-readable medium of claim 241, wherein:

when it is determined that the code corresponding to the first node does not contain the second call to the method of the second node, determining whether the code corresponding to the first node comprises a method having a local variable definition using the second node

when it is determined that the code corresponding to the first node comprises a method having the local variable definition using the second node, displaying a dependency link between the graphical representation of the first node and the graphical representation of the second node.

243. (new) The computer-readable medium of claim 148, wherein the second use of the second node comprises a second local variable definition using the second node in a method of the first node

244. (new) The data processing system of claim 181, wherein the first use comprises a declaration.

245. (new) The data processing system of claim 244, wherein the second use comprises a second declaration.

246. (new) The data processing system of claim 181, wherein the first use of the second node comprises a call to a method of the second node.

247. (new) The data processing system of claim 246, wherein the second use of the second node comprises a second call to the method of the second node.

248. (new) The data processing system of claim 199, wherein the first use and the second use each comprise a declaration.

249. (new) The data processing system of claim 199, wherein the first use and the second use each comprise a call to a method.